

# A COMPARATIVE ANALYSIS OF CONVENTIONAL SOFTWARE DEVELOPMENT APPROACHES VS. FORMAL METHODS IN CALL DISTRIBUTION SYSTEMS

---

MUHAMMAD AJMAL SIDDIQUI<sup>1</sup>, MUHAMMAD SALEEM AKHTER<sup>2</sup>,  
NATASH ALI MIAN<sup>3</sup>

<sup>1</sup>Department of Software Development, TRGWORLD, Lahore  
ajmal\_qau@yahoo.com

<sup>2</sup>Department of Computer Science, Minhaj Univeristy, Lahore  
salch148@hotmail.com

<sup>3</sup>School of Computer and Information Technology, Beaconhouse National University, Lahore  
natash.ali@bnu.edu.pk

**Revised December, 2013**

**ABSTRACT:** *When we think about formal method; the first thing which comes in our mind is mathematical approach. The process of formalization is an approach based on mathematics and used to elaborate the properties of systems (hardware and software). The mathematical modeling or formal methods provide us a framework for large and complex systems. Thus these systems can be specified, analyzed, designed, and verified in a systematic way rather than the approaches which are used conventionally. Formal verification and the methods are applied using theoretical computer science fundamentals to solve the complex and difficult problems in large and complex software and hardware systems to ensure the systems will not fail with run-time errors. Conventional approaches of software verification in call distribution systems rely on quality assurance to verify the system behavior and robustness. The process of software testing cannot show the absence of errors it can only show the presence of errors in software systems. [1] In contrast, the mathematically-based techniques of verification are based on formal methods to prove certain software attributes, for example proving that software does or does not contain the occurrence of errors at run-time such as overflows, divide-by-zero, and access violation, invalid memory access and stack/heap corruption. [1] In this paper later we will have comparative analysis of formal methods vs. conventional software development approaches in call distribution systems. Using this comparison we'll try to identify the methodologies and approaches which would be better in SDLC for call distribution systems.*

**Keywords:** Conventional Approaches;SDLC;Call Distribution Systems;Formal Verification; Formal Methods; Software verification.

**1. Introduction:** In comparison with conventional development methods in call distribution systems; the advantage is that the approaches allow us for a unified transition from requirement engineering, analysis to design and implementation. This transition process plays a major advancement role in software engineering practices but quality of specification when we say in terms of correctness and robustness still remain very critical from quality control point of view. Formal methods provide a logical reasoning (in a mathematical sense) about the properties of a system. Eventually, someone may be able to prove the uncertainty and irregularity in the system behavior; for example deadlocks in multithreaded applications, must not occur. Hence the usage of formal methods in object orientation or structural programming would be much beneficial. [1] The use of formal methods is tightly coupled with static code analysis. The process of diagnosing runtime errors is made through the code verification process which is performed through the source code analysis. [1]

[3] Static code analysis and formal methods together helps in run time error detection and also proving the absence of runtime errors. [1] [2]

We are not 100% guaranteed that formal methods can provide us absolute correctness and robustness however the mathematical modeling ensures at certain level to provide a bug free system. There are many examples of system failures and crashes due to errors in formal specifications. On the other side we have many systems failures due to bugs in conventional software developments approaches as well as in quality assurance of those systems. When we talk about call distribution system; the millions of calls are being routed in a unit time. The failure can cause a lot of financial loss. Though the systems are implemented under very strong SDL's and quality assurance but a little error may cause huge financial loss and companies' trust in business process outsourcing systems.

**2. Implication in call distribution systems:** The use of formal methods in the field of software development is not yet well established. One reason is that the software structure (and hence the model we derive from it) will be far more complex than today's computers can process. So the main disadvantage is that most models are costly. However, by applying some levels of abstractions and localizing our models to small parts of our programs we can check the existence of deadlocks and/or security flaws in communication of some reactive components within the software; which assures if we have secure and deadlock-free interactions between the different components of our design.

Formal methods are applied in high end systems like:

- ✓ High integrity communication protocols
- ✓ Avionics systems i.e. Air Traffic Control systems
- ✓ Life safety and critical systems – Medical related systems
- ✓ Call distribution for high volume Business Process Outsourcing systems

With the study of above mentioned points and the short comparison, we would like to make efforts for true and critical analysis of conventional software development and the formal methods applied to the development of call distribution systems in contact centers and other telecom areas.

AVAYA, CISCO, TRGWORLD, GENESIS and IBM are the main groups having their systems in production and also developing such kind of mega systems for various telecom and BPO areas.

In current era TRG and IBM differs with their competitors as they have introduced new technologies in call routing system. They have introduced is technology is personality based matching in call distribution and call routing through demographic data.

The SDLC is followed with conventional approaches which designing and developing all these systems. Moving towards formal methods we may consider the main area in call distribution system either in switch level or application level:

- ✓ Call routing strategies scripting areas
- ✓ Call routing software applications

In these systems the varieties of strategies are adopted on the basis of business need. We divide the needs in mainly two parts a) Inbound b) Outbound. The common strategies are skill based and least idle. Direct routing is also mainly used strategy in case of inbound routing. Progressive, adaptive and predictive algorithms are also used in Outside Broad cast dialing. [4][5]

All the call routing strategies are designed in system scripting area using interpreted base languages in the system dial plans. In common practices formal methods or any kind of mathematical models are not applied while design and implementing the dial plan. But focus of our paper is on Personality based call routing strategy.

**3. What are personality-based call matching technologies?** Personality based call matching technologies are introduced in this era to improve contact center performance. The two big names TRG and IBM own such an innovative solution.

This innovative solution works how the people communicate and allows call centers to gain the actual benefits of quality interactions e.g. revenue improvements, customer experience encampments and decrements in costs at enterprise level.

TRG launched its innovative solution (SATMAP) with primary goals of developing a transformative technology for contact centers to improve their performances. SATMAP - A TRG proprietary technology was engineered during two years research and development. The SATMAP team of 200 people including software engineers, artificial intelligence scientists, quantitative analysts and statisticians. [9]

In the world of contact centers, the best call center executives have realized that outcome of every call is actually the resultant of conversation between two persons or two representatives; whether the objective of that conversation is to win the customer satisfaction, reduction of costs or to increase the sales.

SATMAP's primary focus is on how we communicate. When two personalities are compatible then there is high probability of effective communication between them. [9]

In personality based matching technology the agents interact in a better way with the caller (in case of and inbound) and callee (in case of outbound). The conversation will end with probability of success with in very short time, saving company and customer time which is in fact real money of the both parties.

**4. How do personality-based call matching technologies work?** Using artificial intelligence algorithms, Personality based technologies like SATMAP perform millions calculations in one second to provide personality matching call pairing and the call pairing based on some other variable using the demographic data.

SATMAP identifies patterns in interactions between agent and the caller. This drives a profitable outcome in this subtle interaction. The artificial intelligence engine of SATMAP analyzes the patterns of hundreds of agents and the caller personality characteristics. These real time pairing decisions are made in less than 0.1 second. [9]

#### **1. Personality Profile of Agent**

Agent personality profile is established when SATMAP is deployed first time. Personality attributes of hundreds of agents are captured in a 20 minute survey. These attributes are stored in Agent Information Management System. It takes two to three weeks of initial survey after the deployment of Agent Information Management System. [9]

#### **2. Indexing of Caller ID against the Customer Database**

In order to determine call pairings; the SATMAP does not depend upon the customers sensitive and proprietary information. It captures the psychographic and demographic attributes from census archives, social media catalogs and commercial data base which are publicly available. All these attributes are indexed with caller ID against customer database. [9]

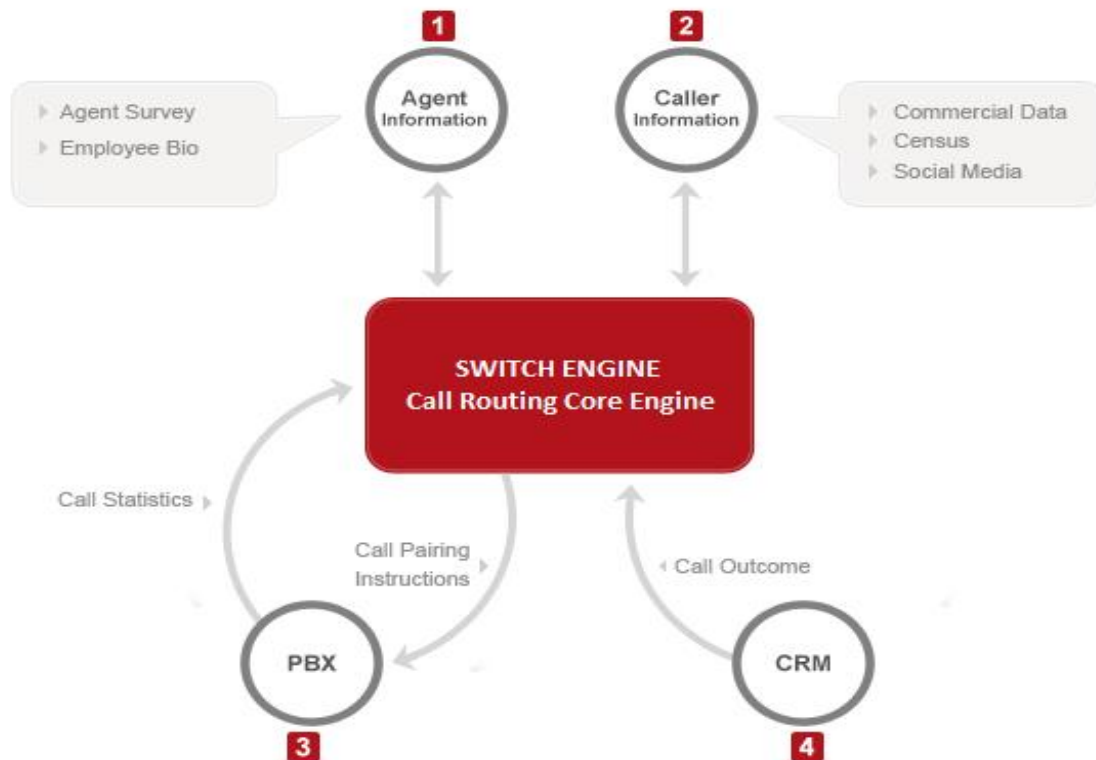
#### **3. Optimization in call pairing decision**

On arriving a call, SATMAP fetches customer and agent attributes information and then start the evaluation process that it completes in a very optimal way in terms of time and system processing. The ultimate selection depends upon the outcomes of previous calls, which is made through the analysis of tens of thousands calls. This outcome is used to identify the

personality based pairings resulting the success of a call with highest probability. All these decisions are made with in few milliseconds. Also the instructions of such a call pairing are passed to host switch through switch interfacing module of SATMAP. [9]

#### 4. Adaptive Learning Mechanism

SATMAP has adaptive learning mechanism in it. It has a built-in mechanism of feedback having capability to improve call pairings with the passage of time. As the system has its core evaluation and learning engine in artificial intelligence so it learns through examination process based on daily call out comes and the call statistics. The daily call volume and the outcome works together to provide the markers personality matching. The improvement in futures results is continues process as SATMAP Engine based on neural network and artificial intelligences analyses the call interactions by feeding call patterns. [9]



**5. Why do we need formal methods call in distribution systems?** BPO call distribution system are designed and developed under traditional software development methodologies. During the analysis, design and development of such system, the traditional software development life cycle is followed, mostly object oriented.

The object oriented analysis and design in convention software development has become an advance paradigm for call routing core component construction. Formal methods provide logical reasoning which is based on mathematical sense about properties of such a software systems.

It has been realized that the usage of formal methods in conventional software development approaches specified in object oriented design, can be much beneficial.

Though, it is clear the object oriented and analysis methodologies provide necessary and sufficient concepts and tools to enhance the quality of call distribution systems but generally they all are informal. This is astonishing as the modeling techniques in object oriented analysis and design have a high potential for formalization. [1] The goal of this study is to make a comparative analysis in formal specification techniques and object oriented analysis and programming in CALL MATCHING/DISTRIBUTION TECHNOLOGY SYSTEMS in the world of contact centers and telecom. In particular, the formalization techniques and the

verification processes in formal methods are mostly discussed and evaluated from software quality assurance point of view. [1]

An objective of assessment of methods to the level of formality is not an easy task. The first step in the elimination of subjectivity is the definition of evaluation criteria. As it is explained in the introduction section, the methods will be evaluated from the quality control point of view.

The specification of quality is defined as the internal correctness and consistency.

**Internal consistency:** We equally emphasis on the description of modeling aspects in conventions software development using object oriented modeling. The static, dynamic and interaction aspects are also equally focused. In this paper, the methods considered, all use different techniques for each modeling aspect. Even the different aspects of objects are modeled using these techniques, might be semantics overlapping exists if we model the same universe. As consequences; for internal consistency, the specification must be checked. [1][3]

**Correctness:** If we use the finite state machines to model the behavior an object types then the exactable system are set of concurrent finite state machines. In the domain of validation, the concurrent state machines are well known specification techniques. These techniques are used to check the protocols for fairness and to avoid system dead locks. In conventional software development, the state machines are used in quite different way. The correctness checking algorithms cannot be transferred to object oriented design in very simple way. However this does not mean such a task is impossible which may define such algorithms which may check conventional software development specification for correctness. [1] [3]

The prerequisite of formal correctness and consistency checking is semantics and syntax of those concepts which are employed by a specific method and also which are unambiguously and rigorously defined. We evaluate them with the following criteria's:

**Criterion 1: Syntax.** The syntax must be defined in a rigorous way and it must not be described loosely [1]

**Criterion 2: Semantics.** Each concept of a method must be provided the format semantics. The concepts are not paraphrased in an informal language or a natural language. Many of the conventional approaches are superficial about the semantics and syntax, also including the concepts which are used by them. So we are compromising on quality if such a specification is made through these methods. Here two examples can be given to elaborate this prospective, one example from data modeling and other from process modeling. The examples illustrate the importance of semantics and precise syntax. [1]

**Criterion 3: Consistency between schemas.** Generally, when the objects are modeled using different techniques, always the emphasis is on static, dynamic and interaction aspects. Nevertheless, the schemas which result are likely to be related. The relationship between static and dynamic interaction schemas must be explicit and to check their consistency. This criterion looks whether a particular method defines formal procedures to check consistency among the subschemas. [1][3]

**Criterion 4: Overall system behavior.** It is only possible to derive the overall system behavior from individual schemas when the consistency among these schemas is established. More particularly, the possibility must exist to compose the behavior of an individual object behavior and also the interaction descriptions into a single system behavior specification. [1] [3]

**Criterion 5: Anomalous system behavior.** If overall system behavior is properly specified, we may check it for desirable properties, e.g. Deadlocks freedom etc. [1] [3]

**6. Modeling Approaches: Data modeling:** In data modeling mostly methods grasp the concept of modeling from EER model with the addition of few proprietary concepts. Usually these proprietary concepts are not defined in formal way. [1] [7]

**Process modeling:** In process modeling, all methods use regular languages and these regular languages are represented either by finite state machines, regular expressions and Harel state charts. Especially when we talk call distribution system, all the dial plans in PBX are designed using automata techniques.

Apparently, finite state machines do not always have starting state and/or an ending state. The vagueness with which the technique of finite state machines are used in object oriented analysis and design seems to be a consequence of the fact that the authors of this method have not given a precise definition of finite state machines. [1] [7]

**Interaction modeling:** In interaction modeling a variety of techniques are used, message passing concept is one of them used in interaction modeling. The semantics of object interaction is not precisely defined except for O/B and Fusion. [1] [7]

**7. Conventional software development approaches - Problems and Solutions:** In traditional approaches; we usually adopt the software engineering practices, methods, and tools in building and maintaining the software. The software engineering principles and practices have changed many conventional aspects and views and producing the software systems. The software quality must be ensured not only in the testing of the software systems but also in the quality of development processes. Requirement engineering process and quality design is more important the software implementation (software programming). In abstract, the conventional software development process is a software engineering process based on requirement engineering and software development life cycle.

In conventional software development life cycle, we have analysis of requirements and specifications. The specification is study to document and discover the exact requirements for the software system to be developed. The design is the activity of constructing the system to meet the requirements. The implementation is the phase where the specification of design is transformed into a program. The programs are implemented in specific high level languages like C, C++, JAVA or C#. Similar concepts and software life cycle paradigms are adopted while constructing the software for call distribution in contact centers. Testing is the last item to detect faults, errors and bugs in programs. There are number of testing techniques applied with variations of different test case to make the system error free. Software release delivery to customer for operation. Release maintenance is an umbrella activity and is modified to fix the existing bugs or to enhance the system with new requirements.

What are the problems with conventional approaches in SDLC? We may answer to the question; the overall problem in software systems is that they don't operate as per the expectations. The software systems usually contain faults or bugs, in other words, we can say software never completes. In general, the faults (bugs/errors) are roughly categorized into three ways:

- ✓ The developed system does not meet the client requirements.
- ✓ Incorrect or bad design (e.g., architecture problem)
- ✓ Bugs in Implementation (e.g., type, array boundary, and file operations).

As an example in software quality problem; In Japan, the National Aerospace Laboratory conducted airplane flight. This was test flight of a supersonic experimental airplane (The rocket powered experimental plane NEXST-1) in Woomera. Australia was part of that Next generation supersonic aircraft Technology Research and Development; unfortunately it was failed due to error in software. [3]

One of the common problems observed in conventional software development approach is behind schedule and over budget. Usually the software requirements are not properly understood before starting the implementation of system. Clients have wake idea of that system that they want to acquire and they don't deliver their requirements properly and precisely. That's why the developers often misinterpret the user requirements. Due to these concerns the software development process is not controlled in better way. The obligations of system designers, analysts and programmers are not defined precisely. The testing process starts very late and also it has very limited power to ensure the correctness of system as testing process can only show the presence of bug not the absence of bugs.

The overall cost of testing process is almost 60% or more of the total cost of a software development.

In life critical software, we may result the loss of life if the systems don't operate properly according to requirement specifications. A very important aspect that can be the reason for these problems; the requirements specifications are written in natural language or some informal languages or in semi-formal notations like DFD or UML. Such kinds of specifications are ambiguous and don't have any foundation to transform them into automated formalization and verification.

**8. Possible benefits in formal methods vs conventional approaches:** Formal methods consist of system formal specification and formal verification. For that mathematical domain set theory, algebra (i.e linear , relational), topology, metric spaces and vector spaces are used. In formal methods we also understand the formal notations (e.g. Z, Petri-nets, and VDM etc.), logical calculus for formal specification and the methods for developing the software system. [8] [2][3]

All these things ensure at good enough level to make the requirements and their development unambiguous as compared the tradition approaches.

We can say this process is a road map from abstract towards the concreteness. The concreteness means here doesn't mean we have the system 100% ensured and bug free. This is relative term not the absolute statement. In this we implement a refinement and verification process between our requirement specification and system implementation. This process continues until we are ensured the system is completely bug free but not exactly 100%.

**9. Problems with formal methods:** In large systems the formal specifications are difficult to read, write and understand for software engineers. Formal methods are unable to integrate properly into the traditional software development process. For example it is difficult to use DFD or a flowchart with formal notations. [3]

Another important aspect, the formal methods are not much effective to model the dynamic properties of software. Graphical use interface, system efficiency are the main examples of this aspect. Formal proof is much difficult to conduct by software engineers because of its cost which is usually very high in terms of labor, time. In system validation the formal proof is also not effective. [3]

**10. Conclusion and future work:** Conventional software development approaches and other programming paradigms can be beneficial in many ways from the availability of mathematical semantics for example consistency and error checking. We have found that most of the time specification techniques written for object orientation are not formal.

Also one important aspect in terms of methods in conventional software development approaches, even not a single method can be reviewed to incorporate the concept of overall system behavior. From software quality point of view in conventional software development process in call distribution systems, the formalization of conventional approaches will not allow the checking of correctness at an earlier stage in conventional software development process, hereby reducing development costs. Formal methods provide the better correctness and the process of run time error checking but this may increase the system cost dramatically for small systems (e.g. Asterisks based systems). So we can apply these methodologies in the development of larger systems (e.g. AVAYA, CISCO, GENESIS, SATMAP, NORTEL, ROCKWEL) would be more beneficial as compared with conventional approaches.

The conclusion is not final but we may work further towards the actual implementations using Formal languages like Petri-nets, Z, and VDM etc. So this conclusion does not end with final results of applying formal methods in call distributed systems.

## REFERENCES

- [1] Snoeck, M., Wijssen, J., & Dedene, G. (1996). Formal specification techniques in object-oriented analysis: a comparative view. DTEW Research Report 9630, 1-16.
- [2] Groote, J. F., Osaiweran, A. A. H., & Wesselius, J. (2011). Benefits of applying formal methods to industrial control software. Computer Science Report, 11(04).
- [3] Gabbar, H. A. (Ed.). (2006). Modern formal methods and applications. Springer Science & Business Media.
- [4] Phithakkitnukoon, Santi; Dantu, Ram; Claxton, Rob and Eagle, Nathan (2011). "Behavior based adaptive call predictor. ACM Transactions on Autonomous and Adaptive Systems"
- [5] Mehrotra, V., Ross, K., Ryder, G., & Zhou, Y. P. (2012). Routing to manage resolution and waiting time in call centers with heterogeneous servers. Manufacturing & service operations management, 14(1), 66-81.
- [6] Coad, P., and Yourdon, E. Object-Oriented analysis. In Prentice Hall, Englewood Cliffs, N.J., 1991.
- [7] Plat, N., van Katwijk, J., & Toetenel, H. (1992). Application and benefits of formal methods in software development. Software Engineering Journal, 7(5), 335-346.
- [8] Zhang, J., & Cheng, B. H. (2006, May). Model-based development of dynamically adaptive software. In Proceedings of the 28th international conference on Software engineering (pp. 371-380). ACM.